

## BAB II TINJAUAN PUSTAKA

### A. Penelitian Terkait

Terdapat beberapa penelitian terkait pengembangan *backend* API, infrastruktur CPS, dan optimasi sistem audio masjid, Tabel 2.1 menunjukkan rangkuman penelitian terdahulu yang digunakan dalam penelitian.

Tabel 2. 1 Ringkasan Penelitian Terdahulu

| No  | Penulis (Tahun)           | Judul                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                       | (3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 1   | Rosad, S., et al. (2025)  | <p><b>Judul:</b> <i>Zoned Audio Distribution: Systematic Review and Optimization Framework in Mosque</i></p> <p><b>Metode:</b> <i>Systematic Literature Review</i> (2003-2025) dan survei empiris pada 77 masjid</p> <p><b>Tools:</b> Perangkat lunak simulasi akustik (ODEON 15.06, Ease Focus 3.0)</p> <p><b>Hasil:</b> Pengembangan kerangka kerja optimasi distribusi audio menggunakan zonasi spasial-akustik untuk meningkatkan <i>Speech Transmission Index</i> (STI) di atas 0.6 dan mengatasi titik mati akustik (<i>dead zones</i>).</p> |
| 2   | Rosad, S., et al. (2025)  | <p><b>Judul:</b> <i>Edge-Based IoT Endpoint Performance Optimization for Smart Mosques</i></p> <p><b>Metode:</b> Eksperimental dengan teknik <i>edge computing</i> (<i>local caching, decoupled computation, push communication</i>)</p> <p><b>Tools:</b> ESP32, Python (Flask)/Node.js, SSE, WebSocket, Raspberry Pi 4</p> <p><b>Hasil:</b> Penurunan latensi median hingga 72% (dari 180ms ke 50ms) dan peningkatan ketersediaan layanan hingga 99,3% untuk kontrol perangkat masjid yang kritis,.</p>                                           |
| 3   | Hamzah, M., et al. (2023) | <p><b>Judul:</b> <i>Distributed Control of Cyber Physical System on Various Domains: A Critical Review</i></p> <p><b>Metode:</b> <i>Critical Review</i> terhadap 14 domain aplikasi CPS</p> <p><b>Hasil:</b> Pemetaan integrasi 3C (<i>Computing, Communication, Control</i>) dalam kerangka CPS terdistribusi yang menjadi landasan teoretis orkestrasi siber-fisik,.</p>                                                                                                                                                                         |

|   |                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 | Pudoli, A., et al. (2023)        | <p><b>Judul:</b> Implementasi Web Service Restful Dengan <i>Autentikasi</i> JSON Web Token Berbasis Web dan Android</p> <p><b>Metode:</b> Rancang bangun sistem dengan arsitektur REST</p> <p><b>Tools:</b> Node.js (Express.js), MySQL, JSON Web Token (JWT), Postman</p> <p><b>Hasil:</b> Berhasil mengimplementasikan sistem akses terpusat yang aman untuk multi-platform menggunakan autentikasi JWT pada <i>web service</i>,.</p> |
| 5 | Umam, M. S., et al. (2023)       | <p><b>Judul:</b> Implementasi Protokol MQTT pada Aplikasi Smart Garden Berbasis IoT</p> <p><b>Metode:</b> Eksperimental pengiriman data sensor melalui jaringan WiFi</p> <p><b>Tools:</b> Protokol MQTT, Mosquitto Broker, Mikrokontroler</p> <p><b>Hasil:</b> Broker MQTT mampu menangkap data sensor dengan sangat akurat dan efisien karena memiliki <i>header</i> yang jauh lebih ringan dibanding HTTP.</p>                        |
| 6 | Tkáčik, M., et al. (2023)        | <p><b>Judul:</b> <i>Modeling and Analysis of Distributed Control Systems: Proposal of a Methodology</i></p> <p><b>Metode:</b> Pemodelan sistem kontrol terdistribusi (DCS) menggunakan <i>Petri Nets</i></p> <p><b>Hasil:</b> Metodologi untuk memprediksi <i>throughput</i> dan waktu respon data sebelum implementasi fisik sistem kontrol terdistribusi berskala besar.,</p>                                                         |
| 7 | Jati, B. P., et al. (2024)       | <p><b>Judul:</b> Literasi Sistem Audio melalui Revitalisasi, Pembenahan dan Optimisasi Audio di Masjid At-Taqwa</p> <p><b>Metode:</b> Revitalisasi fisik dan penataan sistem audio</p> <p><b>Tools:</b> Software Ease Focus, penataan orientasi speaker</p> <p><b>Hasil:</b> Peningkatan audibilitas suara Imam dan kejelasan khutbah melalui optimasi letak, ketinggian, dan kemiringan speaker.,</p>                                  |
| 8 | Sulistiyan, S. R., et al. (2021) | <p><b>Judul:</b> <i>Monitoring and Control System with a Client-Server Model Based on IoT</i></p> <p><b>Metode:</b> Perbandingan performa sistem <i>offline</i> vs <i>online</i></p> <p><b>Tools:</b> NodeMCU ESP-12E, Firebase Cloud, Android</p> <p><b>Hasil:</b> Sistem lokal (<i>offline</i>) terbukti memiliki waktu respon yang jauh lebih cepat dibandingkan sistem berbasis cloud (Firebase) yang rentan latensi internet.</p>  |
| 9 | Nurhayati, E., et al. (2023)     | <p><b>Judul:</b> Rancang Bangun Back-end API pada Aplikasi Mobile AyamHub Menggunakan Framework Node JS Express</p>                                                                                                                                                                                                                                                                                                                     |

|    |                                   |                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                   | <p><b>Metode:</b> Metodologi Waterfall</p> <p><b>Tools:</b> Node.js, Express.js, Google Cloud Platform (GCP), MySQL</p> <p><b>Hasil:</b> Penciptaan infrastruktur server <i>backend</i> yang stabil dan responsif dengan teknik <i>non-blocking</i> untuk menangani banyak permintaan secara simultan,.</p>                                                                                                                  |
| 10 | Olivia, M. (2025)                 | <p><b>Judul:</b> Implementasi Dan Evaluasi REST API Dalam Integrasi Sistem Lintas Platform</p> <p><b>Metode:</b> Pengembangan sistem modular terintegrasi</p> <p><b>Tools:</b> Express.js, Next.js, RESTful API</p> <p><b>Hasil:</b> RESTful API menyediakan solusi fleksibel dan skalabel untuk pengelolaan data <i>real-time</i> antar platform yang berbeda.</p>                                                          |
| 11 | Mukhlash, I., et al. (2025)       | <p><b>Judul:</b> Pengembangan <i>Backend</i> Sistem Dashboard Berbasis IoT Dan Manajemen Distribusi Vertical Crab House</p> <p><b>Metode:</b> Extreme Programming (XP) secara iteratif</p> <p><b>Tools:</b> Laravel, Firebase, MySQL</p> <p><b>Hasil:</b> <i>Backend</i> berhasil memproses data sensor IoT dari ESP32 untuk monitoring dan pengelolaan data yang lebih kompleks di sisi server,.</p>                        |
| 12 | Wicaksono, N., & David, F. (2025) | <p><b>Judul:</b> Pengembangan RESTful API untuk Produk E-Commerce Ritel Dengan Pendekatan Microservices dan Spring Boot</p> <p><b>Metode:</b> Research and Development (R&amp;D)</p> <p><b>Tools:</b> Java Spring Boot, MongoDB, PostgreSQL, MySQL</p> <p><b>Hasil:</b> Sistem mampu merespons 500 transaksi secara stabil dengan <i>throughput</i> tinggi melalui pemisahan layanan independen (<i>microservices</i>),.</p> |
| 13 | Lapi, I. Z., et al. (2023)        | <p><b>Judul:</b> Rancang Bangun Purwarupa Aplikasi Masjid Pintar Menggunakan Platform Kodular Berbasis Internet di Masjid</p> <p><b>Metode:</b> Rancang bangun aplikasi monitoring berbasis IoT</p> <p><b>Tools:</b> NodeMCU ESP8266, Kodular, Firebase,</p> <p><b>Hasil:</b> Purwarupa berhasil melakukan kontrol lampu dan pemantauan intensitas cahaya masjid secara <i>real-time</i> melalui aplikasi mobile,.</p>       |
| 14 | Sumudra, D. G., et al. (2024)     | <p><b>Judul:</b> Rancang Bangun Website Terpadu Menggunakan Framework ReactJS Untuk Masjid Jami' At-Taqwa</p> <p><b>Metode:</b> SDLC model Waterfall</p>                                                                                                                                                                                                                                                                     |

|    |                                       |                                                                                                                                                                                                                                                                                                                                                                                   |
|----|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                       | <b>Tools:</b> ReactJS, Node.js, MySQL<br><b>Hasil:</b> Peningkatan efisiensi manajemen administratif masjid sebesar 91,75% melalui sistem yang terintegrasi dan responsif,.                                                                                                                                                                                                       |
| 15 | Bakti, V. K., et al. (2023)           | <b>Judul:</b> Sistem Monitoring Ruang Data Center Kombinasi Multi Sensor dengan API Tuya<br><b>Metode:</b> Analisis integrasi API pihak ketiga<br><b>Tools:</b> Tuya IoT Development Platform, PHP, WiFi Sensor<br><b>Hasil:</b> Akuisisi data dari berbagai sensor fisik berhasil dilakukan secara daring dengan durabilitas sistem yang baik untuk monitoring ruangan kritis,.  |
| 16 | Muhammad, I. R. D., et al. (2024)     | <b>Judul:</b> Pengembangan <i>Backend</i> Server Berbasis Arsitektur REST API pada Sistem Transfer Dompet Digital<br><b>Metode:</b> Perancangan model data ERD dan UM<br><b>Tools:</b> NestJS, NestCLI, MySQL<br><b>Hasil:</b> Berhasil mengimplementasikan interoperabilitas antar platform yang efisien dengan manajemen data terstruktur menggunakan format JSON,.             |
| 17 | Enda, D., et al. (2021)               | <b>Judul:</b> Rancang Bangun Aplikasi AIS <i>Backend</i> Untuk Pemantauan Lalu Lintas Kapal<br><b>Metode:</b> Prosedur penelitian sistematis (Studi literatur ke pengujian)<br><b>Tools:</b> Framework Flask, Database Relasional<br><b>Hasil:</b> Aplikasi mampu memproses data monitoring secara <i>real-time</i> dengan efisiensi penyimpanan data yang terkelola dengan baik. |
| 18 | Firdaus, A., & Ramadhan, D. A. (2021) | <b>Judul:</b> Pengembangan Back End Berbasis REST API pada Sistem E-Partisipasi Patriot Pangan<br><b>Metode:</b> Metodologi Scrum<br><b>Tools:</b> Node.js, Express.js, Socket.IO<br><b>Hasil:</b> <i>Backend</i> REST API berhasil mendukung notifikasi <i>real-time</i> dan pengelolaan data berjenjang dari level desa hingga provinsi.                                        |
| 19 | Herdiyatmoko, H. F. (2022)            | <b>Judul:</b> Desain Sistem <i>Backend</i> Berbasis REST API Menggunakan Framework Laravel 7<br><b>Metode:</b> Implementasi arsitektur Model-View-Controller (MVC)<br><b>Tools:</b> Laravel 7, Postman, MySQL                                                                                                                                                                     |

|    |                             |                                                                                                                                                                                                                                                                                                                                                                                                     |
|----|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                             | <b>Hasil:</b> Desain <i>backend</i> yang aman dengan empat <i>endpoint</i> API utama untuk operasi CRUD data yang mendukung interoperabilitas perangkat mobile,.                                                                                                                                                                                                                                    |
| 20 | Talha, M. A., et al. (2023) | <b>Judul:</b> <i>IoT-Based Real-Time Monitoring And Control System for Distribution Substation</i><br><b>Metode:</b> Pengembangan sistem monitoring proaktif terhadap kegagalan perangkat<br><b>Tools:</b> Aplikasi Blynk, Sensor IoT<br><b>Hasil:</b> Sistem terbukti efisien dalam mendeteksi anomali suhu dan tegangan secara <i>real-time</i> untuk meningkatkan keandalan infrastruktur fisik. |

Berikut adalah tabel gap penelitian dan Novelty yang digunakan dalam penelitian.

Tabel 2. 2 Rangkuman Gap Penelitian dan Novelty

| No  | Penulis<br>(Tahun)         | Gap Penelitian                                                                                                                                         | Novelty                                                                                                                                                |
|-----|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                        | (3)                                                                                                                                                    | (4)                                                                                                                                                    |
| 1   | Rosad, S., et al. (2025)   | Berfokus pada kerangka kerja zonasi secara akustik dan simulasi statis; belum menyentuh infrastruktur <i>backend</i> untuk manajemen data operasional. | Mengembangkan <i>backend</i> API sebagai pusat manajemen data zona dan simpul audio untuk mendukung implementasi fisik sistem zonasi masjid            |
| 2   | Rosad, S., et al. (2025)   | Berfokus pada optimasi performa <i>edge node</i> (ESP32) untuk menekan latensi; belum menyediakan sistem manajemen data persisten dan riwayat kontrol. | Menyediakan manajemen riwayat tindakan kontrol dan sistem peringatan ( <i>alerts</i> ) pada level <i>backend</i> untuk transparansi operasional sistem |
| 3   | Hamzah, M., et al. (2023)  | Memberikan tinjauan kritis CPS secara umum tanpa implementasi <i>backend</i> API yang spesifik.                                                        | Memberikan kontribusi praktis pada domain CPS masjid melalui implementasi <i>backend</i> sebagai lapisan siber yang mengoordinasikan aset fisik audio  |
| 4   | Pudoli, A., et al. (2023)  | Menggunakan autentikasi JWT untuk sistem informasi administratif umum (absensi); tidak melibatkan interaksi dengan perangkat keras IoT.                | Menerapkan keamanan JWT pada API kontrol perangkat fisik (audio masjid) untuk menjamin hak akses terbatas pada instruksi kontrol yang krusial          |
| 5   | Umam, M. S., et al. (2023) | Implementasi MQTT terbatas pada pemantauan pasif ( <i>monitoring</i> ) satu arah tanpa integrasi REST API untuk                                        | Mengintegrasikan protokol MQTT dengan REST API untuk penyaluran perintah kontrol aktif dari server ke                                                  |

|    |                                    |                                                                                                                                                |                                                                                                                                                                      |
|----|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                    | manajemen data yang kompleks                                                                                                                   | banyak node zona secara terorganisir                                                                                                                                 |
| 6  | Tkáčik, M. (2023)                  | Menekankan pada pemodelan teoretis sistem kontrol skala besar; belum ada implementasi praktis untuk bangunan publik                            | Mengimplementasikan sistem kontrol terdistribusi skala lingkungan uji untuk memvalidasi fungsionalitas CRUD dan respon MQTT pada node audio                          |
| 7  | Jati, B. P., et al. (2024)         | Solusi hanya bersifat revitalisasi fisik dan penataan speaker manual; tidak ada keterlibatan sistem monitoring digital.                        | Menciptakan skema pencatatan log pembacaan SPL secara digital dan terpusat untuk memantau efektivitas distribusi suara pasca-revitalisasi                            |
| 8  | Sulistiyanti, S. R., et al. (2021) | Menganalisis latensi sistem monitoring tanpa menyediakan mekanisme kontrol terdistribusi yang terintegrasi dengan basis data relasional.       | Membangun arsitektur <i>backend</i> dengan basis data relasional yang memetakan hubungan antara data zona, node, dan riwayat kontrol secara terstruktur              |
| 9  | Nurhayati, E., et al. (2023)       | Fokus pada infrastruktur server yang stabil untuk manajemen bisnis (AyamHub); tidak ada aspek kontrol perangkat <i>real-time</i> .             | Menciptakan infrastruktur <i>backend</i> yang responsif khusus untuk menangani trafik data sensor SPL dan perintah kontrol MQTT secara simultan                      |
| 10 | Olivia, M. (2025)                  | Mengevaluasi REST API untuk integrasi sistem lintas platform data permainan, bukan untuk sistem siber-fisik (CPS).                             | Mengadopsi prinsip interoperabilitas REST API untuk menghubungkan berbagai komponen CPS (sensor fisik dan logika siber) pada sistem audio masjid                     |
| 11 | Mukhlash, I., et al. (2025)        | <i>Backend</i> dirancang untuk data sensor lingkungan statis (pH, suhu); tidak menangani dinamika kontrol audio multi-zona masjid.             | Merancang logika <i>backend</i> khusus untuk orkestrasi audio, yang membedakan tipe data pembacaan SPL dengan tipe data instruksi kontrol audio                      |
| 12 | Wicaksono, N., et al. (2025)       | Arsitektur <i>microservices</i> yang kompleks mungkin tidak efisien untuk kontrol IoT lokal masjid yang spesifik.                              | Mengoptimalkan <i>backend</i> API monolitik yang tangguh untuk lingkungan uji terbatas, yang mencakup seluruh fungsi kontrol dan monitoring dalam satu ekosistem     |
| 13 | Lapi, I. Z., et al. (2023)         | Purwarupa terbatas pada kontrol lampu dan pembacaan intensitas cahaya melalui platform Kodular yang sangat bergantung pada cloud pihak ketiga. | Mengembangkan API mandiri dengan integrasi MQTT yang lebih fleksibel dan dapat dikelola sendiri tanpa ketergantungan penuh pada platform <i>no-code</i> pihak ketiga |

|    |                                   |                                                                                                                      |                                                                                                                                                          |
|----|-----------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14 | Sumudra, D. G., et al. (2024)     | Website masjid terbatas pada administrasi keuangan dan jadwal sholat; tidak ada fungsi kontrol fisik.                | Memperluas kapabilitas digital masjid dengan integrasi modul kontrol audio yang terhubung langsung ke infrastruktur fisik melalui API                    |
| 15 | Bakti, V. K., et al. (2023)       | Bergantung pada API pihak ketiga (Tuya) untuk akuisisi data sensor; kontrol terbatas pada ekosistem vendor tersebut. | Mengembangkan API kustom berbasis Python yang memberikan kendali penuh atas logika kontrol MQTT tanpa terikat pada ekosistem vendor tertentu             |
| 16 | Muhammad, I. R. D., et al. (2024) | Fokus pada keamanan dan interoperabilitas data finansial (dompet digital)                                            | Mengadaptasi manajemen data JSON yang terstruktur untuk pertukaran informasi antara <i>backend</i> dengan node sensor SPL lapangan                       |
| 17 | Enda, D., et al. (2021)           | Fokus pada efisiensi algoritma penyaringan data posisi kapal; tidak relevan dengan parameter audio.                  | Menerapkan pengolahan data real-time pada <i>backend</i> untuk mendeteksi anomali pada node audio dan memicu peringatan sistem secara otomatis.          |
| 18 | Firdaus, A., et al. (2021)        | Mengembangkan <i>backend</i> untuk sistem partisipasi publik tanpa interaksi dengan perangkat keras lapangan.        | Menggunakan metodologi pengembangan terstruktur untuk memastikan fungsionalitas <i>backend</i> dalam mengelola status aktif/non-aktif banyak node IoT    |
| 19 | Herdiyatomoko, H. F. (2022)       | Desain <i>backend</i> REST API umum menggunakan Laravel untuk data statis.                                           | Menggunakan Python (Fast API) untuk <i>backend</i> yang lebih ringan dalam integrasi dengan pustaka komunikasi kontrol terdistribusi                     |
| 20 | Talha, M. A., et al. (2023)       | Sistem pemantauan gardu listrik yang bersifat proaktif namun tidak memiliki struktur pembagian zona audio.           | Mengadaptasi mekanisme monitoring proaktif ke dalam domain audio masjid untuk mendeteksi kegagalan perangkat audio melalui sistem <i>alert backend</i> . |

Berdasarkan sintesis terhadap berbagai literatur terkait yang disajikan pada Tabel 2, celah penelitian (gap) utama terletak pada kurangnya integrasi sistem kendali siber dengan infrastruktur fisik audio masjid secara terpusat. Riset terdahulu mayoritas masih berfokus pada audit akustik statis, simulasi statis, serta pendekatan berbasis penataan fisik speaker secara manual untuk mencapai audibilitas. Meskipun teknologi IoT mulai diterapkan di masjid, implementasinya masih bersifat parsial dan terbatas pada fungsi monitoring atau kontrol sederhana, serta

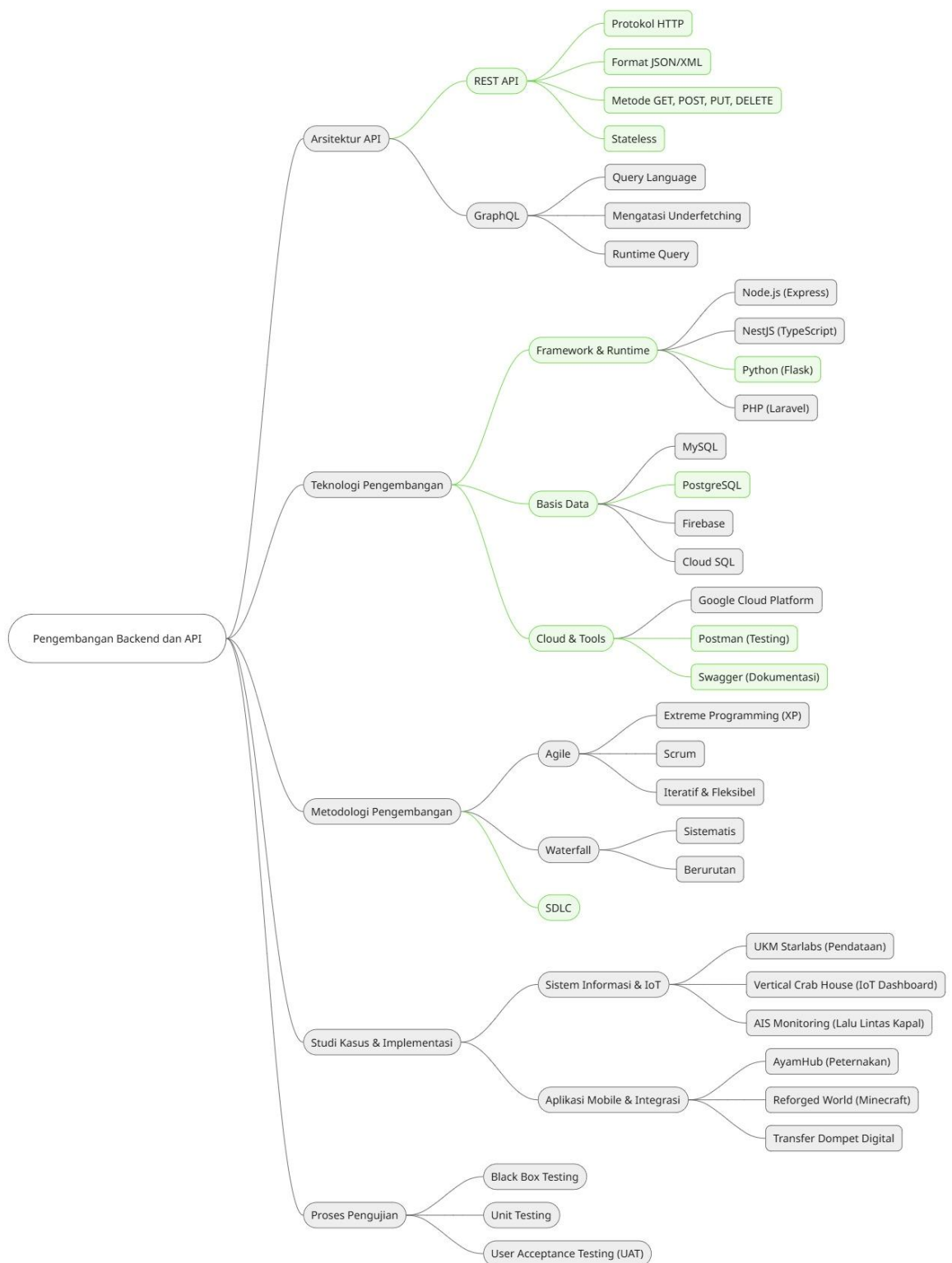
belum mendukung orkestrasi audio multi-zona dalam kerangka CPS yang terintegrasi.

Sebagai solusi atas celah tersebut, kebaruan (novelty) penelitian ini terletak pada pengembangan infrastruktur orkestrasi *backend* API sebagai pusat kendali digital yang menjembatani manajemen zona audio dengan mekanisme kontrol terdistribusi secara near real-time. Sistem ini mengintegrasikan pengelolaan data persisten yang mencakup pembacaan parameter akustik (SPL), riwayat tindakan kontrol, serta sistem peringatan (alerts) otomatis melalui protokol MQTT, serta dilengkapi autentikasi berbasis JWT untuk menjamin akuntabilitas instruksi kontrol terhadap perangkat audio fisik. Dengan demikian, penelitian ini menggeser paradigma optimasi audio masjid dari pendekatan reposisi fisik manual menuju orkestrasi berbasis perangkat lunak yang cerdas dan terintegrasi.

Untuk memperjelas kerangka konseptual penelitian, berikut disajikan mind map penelitian dan pemetaan berdasarkan Taxonomy Penelitian.



Gambar 2. 1 Mind Map Penelitian



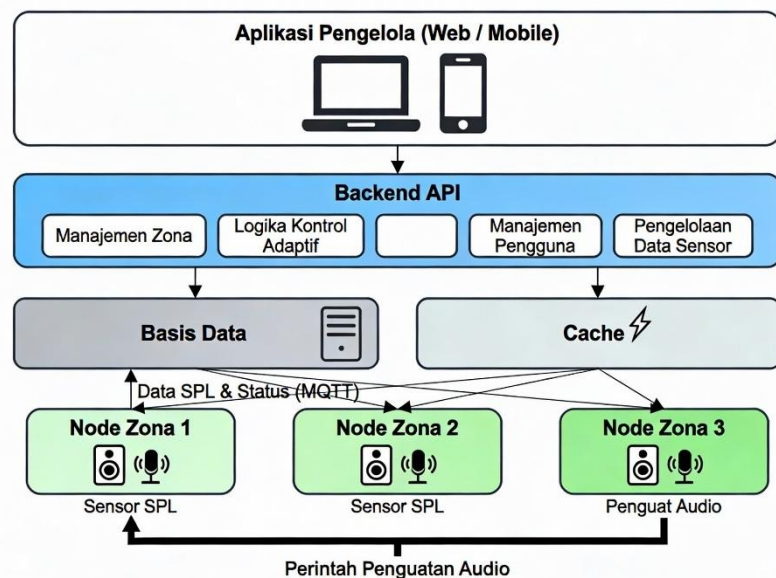
Gambar 2. 2 Taxonomy Penelitian

## B. Landasan Teori

### 1. Sistem Audio Masjid dan CPS

Sistem audio masjid umumnya terdiri atas mikrofon, pengolah sinyal, penguat daya, dan pengeras suara yang didistribusikan di berbagai titik bangunan. Pada banyak masjid, pola yang paling banyak digunakan adalah sistem terpusat (*centralized*) sehingga pengaturan penguatan suara untuk seluruh ruangan dikendalikan dari satu titik utama. Pendekatan ini sering menimbulkan masalah seperti distribusi suara tidak merata, zona yang terlalu keras, dan zona yang terlalu pelan.

Konsep sistem CPS memadukan komponen fisik (sensor, aktuator, perangkat audio) dengan komputasi dan jaringan komunikasi sehingga terjadi siklus umpan balik: mengindra (*sense*), menghitung/mengolah (*compute*), dan menggerakkan (*actuate*)[13]. Dalam konteks masjid, node audio di setiap zona dapat dianggap sebagai bagian fisik yang terhubung ke *backend*, sedangkan *backend* API berperan sebagai pusat pengolahan dan pengambil keputusan[10].



Gambar 2. 3 Ilustrasi Sistem Audio Masjid Berbasis CPS

Secara konseptual, arsitektur sistem audio masjid berbasis CPS dapat digambarkan sebagai berikut.

- Lapisan pengguna: aplikasi web atau gawai untuk pemantauan dan pengaturan zona oleh pengelola.
- Lapisan *backend*: *Backend* API berperan sebagai pusat pengolahan yang mengelola data zona, node, data sensor, serta penerusan perintah kontrol ke node di lapangan.
- Lapisan data: basis data relasional dan penyimpanan cepat (*cache*) untuk data sensor terbaru.

- d. Lapisan lapangan: node-node di tiap zona yang mengukur SPL dan menerapkan perintah penguatan audio.

## 2. Sistem Kontrol Terdistribusi

Sistem kontrol terdistribusi membagi fungsi pengendalian ke beberapa simpul (node) yang saling berkomunikasi melalui jaringan, sehingga setiap node dapat memproses data dan mengambil keputusan secara lokal, berbeda dengan sistem kontrol terpusat yang bergantung pada satu pengendali utama[3]. Arsitektur ini meningkatkan keandalan dan toleransi kesalahan (*fault tolerance*) serta memungkinkan koordinasi antar node secara *real-time* dalam konteks CPS dan IoT[5]. Implementasi sistem terdistribusi dapat ditemukan pada pemantauan dan kendali *real-time* berbasis IoT, seperti gardu distribusi daya, di mana data sensor dikirim, diproses, dan perintah kontrol disalurkan kembali ke node sesuai kondisi operasi[4]. Pendekatan terdistribusi tidak hanya memisahkan fungsi, tetapi juga meningkatkan skalabilitas, efisiensi komunikasi, dan keandalan sistem secara keseluruhan.

## 3. Arsitektur REST API

*Representational State Transfer* (REST) merupakan gaya arsitektur perangkat lunak yang diperkenalkan oleh Roy Fielding dalam disertasinya di *University of California, Irvine* pada tahun 2000[14]. REST mendefinisikan seperangkat prinsip untuk merancang sistem terdistribusi berbasis jaringan yang menekankan pada kesederhanaan, skalabilitas, dan interoperabilitas. Dalam arsitektur REST, setiap sumber daya (*resource*) direpresentasikan melalui *Uniform Resource Identifier* (URI) dan dimanipulasi menggunakan metode standar pada protokol HTTP seperti GET, POST, PUT, dan DELETE[15].

REST API merupakan implementasi *Application Programming Interface* (API) yang menerapkan prinsip-prinsip arsitektur REST dalam pengembangan layanan berbasis web[16]. Salah satu karakteristik utama REST API adalah sifat *stateless*, yaitu setiap permintaan (*request*) dari klien ke server harus memuat seluruh informasi yang diperlukan untuk diproses tanpa bergantung pada status yang disimpan pada server. Pendekatan ini terbukti mendukung skalabilitas sistem dan efisiensi komunikasi pada pengembangan *backend* modern[17]. Selain itu, REST API menerapkan konsep *client-server separation* yang memisahkan tanggung jawab antara antarmuka pengguna dan pengelolaan data, sehingga meningkatkan modularitas serta kemudahan pengembangan sistem[18].

Dalam implementasinya, REST API umumnya menggunakan format representasi data seperti *JavaScript Object Notation* (JSON) karena bersifat ringan dan mudah diproses oleh berbagai platform[19]. Dengan pendekatan

berbasis *resource* dan komunikasi *stateless*, REST API banyak digunakan pada sistem terdistribusi dan aplikasi berbasis web untuk mendukung integrasi antar sistem yang heterogen serta meningkatkan skalabilitas layanan[16], [17].

#### 4. Protokol MQTT

*Message Queuing Telemetry Transport* (MQTT) merupakan protokol komunikasi ringan berbasis *publish–subscribe* yang dirancang untuk perangkat dengan sumber daya terbatas dan jaringan dengan *bandwidth* rendah. MQTT banyak digunakan pada sistem IoT karena efisien dalam pertukaran data serta mendukung komunikasi asinkron melalui perantara yang disebut broker[20], [21].

Dalam arsitektur MQTT, perangkat yang mengirim data disebut *publisher*, sedangkan perangkat yang menerima data disebut *subscriber*. *Publisher* mengirimkan pesan ke broker berdasarkan topik tertentu, dan *subscriber* menerima pesan sesuai topik yang telah didaftarkan. Mekanisme ini memungkinkan komunikasi yang terdistribusi tanpa koneksi langsung antar perangkat serta mendukung implementasi pada berbagai sistem IoT berbasis kontrol dan pemantauan[21], [22].

MQTT mendukung berbagai tingkat kualitas layanan (*Quality of Service/QoS*) untuk menjamin keandalan pengiriman pesan[23], serta memiliki *overhead* komunikasi yang relatif kecil dibandingkan protokol berbasis HTTP[24]. Karakteristik tersebut menjadikan MQTT sesuai untuk sistem pemantauan dan kontrol yang memerlukan pertukaran data secara hampir waktu nyata (*near real-time*) dengan beban komunikasi yang ringan[20], [22].

#### 5. Basis Data Relasional

Basis data relasional merupakan model penyimpanan data yang merepresentasikan informasi dalam bentuk tabel (*relations*) yang terdiri atas baris (*tuples*) dan kolom (*attributes*), di mana setiap tabel memiliki kunci utama (*primary key*) dan dapat saling terhubung melalui kunci asing (*foreign key*) untuk menjaga integritas referensial dan konsistensi data[25]. Model ini banyak digunakan dalam sistem informasi karena kemampuannya menstrukturkan data secara sistematis dan mendukung proses *query* yang efisien.

Perancangan basis data relasional mengikuti prinsip normalisasi, yaitu prosedur untuk mengurangi redundansi data dan mencegah anomali saat melakukan penyisipan, pembaruan, atau penghapusan data. Normalisasi membantu menjaga konsistensi dan kualitas data dalam skala besar, serta memudahkan pengelolaan data historis yang kompleks[26]. Selain itu, penerapan indeks (*indexing*) pada atribut tertentu dapat meningkatkan

performa pencarian dan pemrosesan data, khususnya pada sistem dengan frekuensi pembacaan data yang tinggi.

Dalam konteks sistem berbasis IoT, basis data relasional digunakan untuk menyimpan data konfigurasi, data pengguna, serta data historis hasil pengukuran sensor. Struktur relasional yang terorganisir memungkinkan *backend* melakukan pengolahan data secara terpusat, mendukung pemantauan kondisi sistem, dan memfasilitasi pengambilan keputusan secara sistematis[27].

Selain itu, studi literatur membandingkan model basis data relasional dengan model NoSQL, menunjukkan bahwa relasional unggul dalam menjaga integritas data yang terstruktur dan mendukung aplikasi yang membutuhkan skema tetap serta reliabilitas tinggi[28]. Hal ini menjadikan basis data relasional pilihan tepat untuk sistem *backend* yang mengelola data konfigurasi, riwayat sensor, dan log tindakan kontrol pada sistem audio multi-zona berbasis CPS.

## 6. Python

Python merupakan bahasa pemrograman yang dikembangkan oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. Python termasuk bahasa pemrograman tingkat tinggi (*high-level programming language*) yang bersifat interpreted dan dirancang dengan sintaks yang sederhana sehingga mudah dipelajari serta dikembangkan[29]. Karakteristik tersebut menjadikan Python banyak digunakan dalam berbagai bidang, seperti pengembangan aplikasi web, otomatisasi sistem, analisis data, IoT, dan kecerdasan buatan.

Python mendukung berbagai paradigma pemrograman, seperti pemrograman prosedural, berorientasi objek (*Object-Oriented Programming/OOP*), dan fungsional[29], [30]. Selain itu, Python memiliki pustaka standar (*standard library*) yang lengkap serta didukung oleh manajer paket PIP (*Python Package Installer*) yang memudahkan pengembang dalam menginstal dan mengelola berbagai modul tambahan sesuai kebutuhan aplikasi.

Dalam penelitian ini, Python digunakan sebagai bahasa pemrograman utama pada sisi *backend* untuk mengelola komunikasi data melalui protokol MQTT, menyediakan layanan API, serta melakukan pencatatan data tingkat SPL ke dalam basis data[31]. Pemanfaatan berbagai pustaka Python memungkinkan proses pengembangan sistem dilakukan secara lebih cepat, terstruktur, dan mudah dipelihara.